Learning Multi-Relational Semantics Using Neural-Embedding Models

Bishan Yang[*]
Cornell University
Ithaca, NY, 14850, USA
`bishan@cs.cornell.edu`

Wen-tau Yih, Xiaodong He, Jianfeng Gao, Li Deng
Microsoft Research
Redmond, WA 98052, USA
`scottyih,xiaohe,jfgao,deng@microsoft.com`

Real-world entities (e.g., people and places) are often connected via relations, forming multi-relational data. Modeling multi-relational data is important in many research areas, from natural language processing to biological data mining [6]. Prior work on multi-relational learning can be categorized into three categories: (1) statistical relational learning (SRL) [10], such as Markov-logic networks [21], which directly encode multi-relational graphs using probabilistic models; (2) path ranking methods [16, 7], which explicitly explore the large relational feature space of relations with random walk; and (3) embedding-based models, which embed multi-relational knowledge into low-dimensional representations of entities and relations via tensor/matrix factorization [24, 18, 19], Bayesian clustering framework [15, 26], and neural networks [20, 2, 1, 25]. Our work focuses on the study of neural-embedding models, where the representations are learned in a neural network architecture. They have shown to be powerful tools for multi-relational learning and inference due to their high scalability and strong generalization abilities.

A number of techniques have been recently proposed to learn entity and relation representations using neural networks [2, 1, 25]. They all represent entities as low-dimensional vectors and represent relations as operators that combine the representations of two entities. The main difference among these techniques lies in the parametrization of the relation operators. For instance, given two entity vectors, the model of Neural Tensor Network (NTN) [25] represents each relation as a bilinear tensor operator and a linear matrix operator. The model of TransE [1], on the other hand, represents each relation as a single vector that linearly interacts with the entity vectors. Both models report promising performance on predicting unseen relationships in knowledge bases. However, they have not been directly compared in terms of the different choices of relation operators and of the resulting effectiveness. Neither has the design of entity representations in these recent studies been carefully explored. For example, NTN [25] first shows the benefits of representing entities as an average of word vectors and initializing word vectors with pre-trained vectors from large text corpora. This idea is promising as pre-trained vectors tend to capture syntactic and semantic information from natural language and can assist in better generalization of entity embeddings. However, many real-world entities are expressed as non-compositional phrases (e.g. person names, movie names, etc.), of which meaning cannot be composed by their constituent words. Therefore, averaging word vectors may not provide an appropriate representation for such entities.

In this paper, we examine and compare different types of relation operators and entity vector representations under a general framework for multi-relational learning. Specifically, we derive several recently proposed embedding models, including TransE [1] and NTN [25], and their variants under the same framework. We empirically evaluate their performance on a knowledge base completion task using various real-world datasets in a controlled experimental setting and present several interesting findings. First, the models with fewer parameters tend to be better than more complex models in terms of both performance and scalability. Second, the bilinear operator plays an important role in capturing entity interactions. Third, with the same model complexity, multiplicative operations are superior to additive operations in modeling relations. Finally, initializing entity vectors with pre-trained phrase vectors can significantly boost performance, whereas representing entity vectors as an average of word vectors that are initialized with pre-trained vectors may hurt performance.

[*]Work conducted while interning at Microsoft Research.

These findings have further inspired us to design a simple knowledge base embedding model that significantly outperforms existing models in predicting unseen relationships, with a top-10 accuracy of 73.2% (vs. 54.7% by TransE) evaluated on Freebase.

1 A General Framework for Multi-Relational Representation Learning

Most existing neural embedding models for multi-relational learning can be derived from a general framework. The input is a relation triplet (e_1, r, e_2) describing e_1 (the *subject*) and e_2 (the *object*) are in a certain relation r. The output is a scalar measuring the validity of the relationship. Each input entity can be represented as a high-dimensional sparse vector ("one-hot" index vector or "n-hot" feature vector). The first neural network layer projects the input vectors to low dimensional vectors, and the second layer projects these vectors to a real value for comparison via a relation-specific operator (it can also be viewed as a scoring function).

More formally, denote $\mathbf{x}_{e_i}$ as the input for entity e_i and $\mathbf{W}$ as the first layer neural network parameter. The scoring function for a relation triplet (e_1, r, e_2) can be written as

$$S_{(e_1,r,e_2)} = G_r\big(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}\big), \text{ where } \mathbf{y}_{e_1} = f\big(\mathbf{W}\mathbf{x}_{e_1}\big),\ \ \mathbf{y}_{e_2} = f\big(\mathbf{W}\mathbf{x}_{e_2}\big) \tag{1}$$

Many choices for the form of the scoring function G_r are available. Most of the existing scoring functions in the literature can be unified based on a basic linear transformation g_r^a, a bilinear transformation g_r^b or their combination, where g_r^a and g_r^b are defined as

$$g_r^a(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) = \mathbf{A}_r^T \begin{pmatrix} \mathbf{y}_{e_1} \\ \mathbf{y}_{e_2} \end{pmatrix} \ \text{ and } \ g_r^b(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) = \mathbf{y}_{e_1}^T \mathbf{B}_r \mathbf{y}_{e_2}, \tag{2}$$

which are parametrized by $\mathbf{A}_r$ and $\mathbf{B}_r$, respectively.

In Table 1 we summarize several popular scoring functions in the literature for a relation triplet (e_1, r, e_2), reformulated in terms of the above two functions. Denote by $\mathbf{y}_{e_1}, \mathbf{y}_{e_2} \in R^n$ two entity vectors. Denote by $\mathbf{Q}_{r_1}, \mathbf{Q}_{r_2} \in R^{m\times n}$ and $\mathbf{V}_r \in R^n$ matrix or vector parameters for linear transformation g_r^a . Denote by $\mathbf{M}_r \in R^{n\times n}$ and $\mathbf{T}_r \in R^{n\times n\times m}$ matrix or tensor parameters for bilinear transformation g_r^b. $\mathbf{I} \in R^n$ is an identity matrix. $\mathbf{u}_r \in R^m$ is an additional parameter for relation r. The scoring function for TransE is derived from $||\mathbf{y}_{e_1} - \mathbf{y}_{e_2} + V_r||_2^2$ as in [1].

Models	$\mathbf{B}_r$	$\mathbf{A}_r$	Scoring Function G_r
Distance [3]	-	$\begin{bmatrix}\mathbf{Q}_{r_1} & -\mathbf{Q}_{r_2}\end{bmatrix}$	$-\|\|g_r^a(\mathbf{y}_{e_1}, \mathbf{y}_{e_2})\|\|_1$
Single Layer [25]	-	$\begin{bmatrix}\mathbf{Q}_{r1} & \mathbf{Q}_{r2}\end{bmatrix}$	$\mathbf{u}_r^T \tanh(g_r^a(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}))$
TransE [1]	$\mathbf{I}$	$\begin{bmatrix}\mathbf{V}_r^T & -\mathbf{V}_r^T\end{bmatrix}$	$2g_r^a(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) - 2g_r^b(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) + \|\|\mathbf{V}_r\|\|_2^2$
Bilinear [14]	$\mathbf{M}_r$	-	$g_r^b(\mathbf{y}_{e_1}, \mathbf{y}_{e_2})$
NTN [25]	$\mathbf{T}_r$	$\begin{bmatrix}\mathbf{Q}_{r1} & \mathbf{Q}_{r2}\end{bmatrix}$	$\mathbf{u}_r^T \tanh\big(g_r^a(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) + g_r^b(\mathbf{y}_{e_1}, \mathbf{y}_{e_2})\big)$

Table 1: Comparisons among several multi-relational models in their scoring functions.

This general framework for relationship modeling also applies to the recent deep-structured semantic model [12, 22, 23, 9, 28], which learns the relevance or a single relation between a pair of word sequences. The framework above applies when using multiple neural network layers to project entities and using a relation-independent scoring function $G_r\big(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}\big) = \cos[\mathbf{y}_{e_1}(\mathbf{W}_r), \mathbf{y}_{e_2}(\mathbf{W}_r)]$. The cosine scoring function is a special case of g_r^b with normalized $\mathbf{y}_{e_1}, \mathbf{y}_{e_2}$ and $\mathbf{B}_r = \mathbf{I}$.

The neural network parameters of all the models discussed above can be learned by minimizing a margin-based ranking objective[1], which encourages the scores of positive relationships (triplets) to be higher than the scores of any negative relationships (triplets). Usually only positive triplets are observed in the data. Given a set of positive triplets T, we can construct a set of negative triplets T' by corrupting either one of the relation arguments, $T' = \{(e_1', r, e_2) | e_1' \in E, (e_1', r, e_2) \notin T\} \cup \{(e_1, r, e_2') | e_2' \in E, (e_1, r, e_2') \notin T\}$. The training objective is to minimize the margin-based ranking loss

$$L(\Omega) = \sum_{(e_1,r,e_2)\in T} \sum_{(e_1',r,e_2')\in T'} \max\{S_{(e_1',r,e_2')} - S_{(e_1,r,e_2)} + 1, 0\} \tag{3}$$

[1]Other objectives such as mutual information (as in [12]) and reconstruction loss (as in tensor decomposition approaches [4]) can also be applied. Comparisons among these objectives are beyond the scope of this paper.

2 Experiments and Discussion

Datasets and evaluation metrics We used the WordNet (WN) and Freebase (FB15k) datasets introduced in [1]. WN contains $151,442$ triplets with $40,943$ entities and 18 relations, and FB15k consists of $592,213$ triplets with $14,951$ entities and 1345 relations. We also consider a subset of FB15k (FB15k-401) containing only frequent relations (relations with at least 100 training examples). This results in $560,209$ triplets with $14,541$ entities and 401 relations. We use link prediction as our prediction task as in [1]. For each test triplet, each entity is treated as the target entity to be predicted in turn. Scores are computed for the correct entity and all the corrupted entities in the dictionary and are ranked in descending order. We consider *Mean Reciprocal Rank (MRR)* (an average of the reciprocal rank of an answered entity over all test triplets), *HITS@10* (top-10 accuracy), and *Mean Average Precision* (MAP) (as used in [4]) as the evaluation metrics.

Implementation details All the models were implemented in C# and using GPU. Training was implemented using mini-batch stochastic gradient descent with AdaGrad [8]. At each gradient step, we sampled for each positive triplet two negative triplets, one with a corrupted subject entity and one with a corrupted object entity. The entity vectors are renormalized to have unit length after each gradient step (it is an effective technique that empirically improved all the models). For the relation parameters, we used standard L2 regularization. For all models, we set the number of mini-batches to 10, the dimensionality of the entity vector $d = 100$, the regularization parameter 0.0001, and the number of training epochs $T = 100$ on FB15k and FB15k-401 and $T = 300$ on WN (T determined based on the learning curves where the performance of all models plateaued.) The learning rate was initially set to 0.1 and then adapted over the course of training by AdaGrad.

2.1 Model Comparisons

We examine five embedding models in decreasing order of complexity: (1) NTN with 4 tensor slices as in [25]; (2) Bilinear+Linear, NTN with 1 tensor slice and without the non-linear layer; (3) TransE with L2 norm , a special case of Bilinear+Linear as described in [1]; (4) Bilinear; (5) Bilinear-diag: a special case of Bilinear where the relation matrix is a diagonal matrix.

	FB15k		FB15k-401		WN	
	MRR	HITS@10	MRR	HITS@10	MRR	HITS@10
NTN	0.25	41.4	0.24	40.5	0.53	66.1
Blinear+Linear	0.30	49.0	0.30	49.4	0.87	91.6
TransE (DISTADD)	0.32	53.9	0.32	54.7	0.38	90.9
Bilinear	0.31	51.9	0.32	52.2	**0.89**	92.8
Bilinear-diag (DISTMULT)	**0.35**	**57.7**	**0.36**	**58.5**	0.83	**94.2**

Table 2: Comparison of different embedding models

Table 2 shows the results of all compared methods on all the datasets. In general, we observe that the performance increases as the complexity of the model decreases on FB. NTN, the most complex model, provides the worst performance on both FB and WN, which suggests overfitting. Compared to the previously published results of TransE [1], our implementation achieves much better results (53.9% vs. 47.1% on FB15k and 90.9% vs. 89.2% on WN) using the same evaluation metric (HITS@10). We attribute such discrepancy mainly to the different choice of SGD optimization: AdaGrad vs. constant learning rate. We also found that Bilinear consistently provides comparable or better performance than TransE, especially on WN. Note that WN contains much more entities than FB, it may require the parametrization of relations to be more expressive to better handle the richness of entities. Interestingly, we found that a simple variant of Bilinear – BILINEAR-DIAG, clearly outperforms all baselines on FB and achieves comparable performance to Bilinear on WN.

2.2 Multiplicative vs. Additive Interarctions

Note that BILINEAR-DIAG and TRANSE have the same number of model parameters and their difference lies in the operational choices of the composition of two entity vectors – BILINEAR-DIAG uses weighted element-wise dot product (multiplicative operation) and TRANSE uses element-wise subtraction with a bias (additive operation). To highlight the difference, here we use DISTMULT and

DISTADD to refer to BILINEAR-DIAG and TRANSE, respectively. Comparison between these two models can provide us with more insights on the effect of two common choices of compositional operations – multiplication and addition for modeling entity relations. Overall, we observed superior performance of DISTMULT on all the datasets in Table 2. Table 3 shows the *HITS@10* score on four types of relation categories (as defined in [1]) on FB15k-401 when predicting the subject entity and the object entity respectively. We can see that DISTMULT significantly outperforms DISTADD in almost all the categories. More qualitative results can be found in the Appendix.

	Predicting subject entities				Predicting object entities			
	1-to-1	1-to-n	n-to-1	n-to-n	1-to-1	1-to-n	n-to-1	n-to-n
DISTADD	70.0	76.7	21.1	53.9	68.7	17.4	**83.2**	57.5
DISTMULT	**75.5**	**85.1**	**42.9**	**55.2**	**73.7**	**46.7**	81.0	**58.8**

Table 3: Results by relation categories: one-to-one, one-to-many, many-to-one and many-to-many

2.3 Entity Representations

In the following, we examine the learning of entity representations and introduce two further improvements: using non-linear projection and initializing entity vectors with pre-trained phrase vectors. We focus on DISTMULT as our baseline and compare it with the two modifications DISTMULT-tanh (using $f = \tanh$ for entity projection) and DISTMULT-tanh-EV-init (initializing the entity parameters with the 1000-dimensional pre-trained phrase vectors released by *word2vec* [17]) on FB15k-401. We also reimplemented the word vector representation and initialization technique introduced in [25] – each entity is represented as an average of its word vectors and the word vectors are initialized using the 300-dimensional pre-trained word vectors released by *word2vec*. We denote this method as DISTMULT-tanh-WV-init. Inspired by [4], we design a new evaluation setting where the predicted entities are automatically filtered according to "entity types" (entities that appear as the subjects/objects of a relation have the same type defined by that relation). This provides us with better understanding of the model performance when some entity type information is provided. In

	MRR	HITS@10	MAP (w/ type checking)
DISTMULT	0.36	58.5	64.5
DISTMULT-tanh	0.39	63.3	76.0
DISTMULT-tanh-WV-init	0.28	52.5	65.5
DISTMULT-tanh-EV-init	**0.42**	**73.2**	**88.2**

Table 4: Evaluation with pre-trained vectors

Table 4, we can see that DISTMULT-tanh-EV-init provides the best performance on all the metrics. Surprisingly, we observed performance drops by DISTMULT-tanh-WV-init. We suspect that this is because word vectors are not appropriate for modeling entities described by non-compositional phrases (more than 73% of the entities in FB15k-401 are person names, locations, organizations and films). The promising performance of DISTMULT-tanh-EV-init suggests that the embedding model can greatly benefit from pre-trained entity-level vectors.

3 Conclusion

In this paper we present a unified framework for modeling multi-relational representations, scoring, and learning, and conduct an empirical study of several recent multi-relational embedding models under the framework. We investigate the different choices of relation operators based on linear and bilinear transformations, and also the effects of entity representations by incorporating unsupervised vectors pre-trained on extra textual resources. Our results show several interesting findings, enabling the design of a simple embedding model that achieves the new state-of-the-art performance on a popular knowledge base completion task evaluated on Freebase. Given the recent successes of deep learning in various applications; e.g. [11, 27, 5], our future work will aim to exploit deep structure including possibly tensor construct in computing the neural embedding vectors; e.g. [12, 29, 13]. This will extend the current multi-relational neural embedding model to a deep version that is potentially capable of capturing hierarchical structure hidden in the input data.

References

[1] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*, 2013.

[2] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, pages 1–27, 2013.

[3] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *AAAI*, 2011.

[4] Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Chris Meek. Typed tensor decomposition of knowledge bases for relation extraction. In *EMNLP*, 2014.

[5] Li Deng, G. Hinton, and B. Kingsbury. New types of deep neural network learning for speech recognition and related applications: An overview. In *in ICASSP*, 2013.

[6] Pedro Domingos. Prospects and challenges for multi-relational data mining. *ACM SIGKDD explorations newsletter*, 5(1):80–83, 2003.

[7] Xin Luna Dong, K Murphy, E Gabrilovich, G Heitz, W Horn, N Lao, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *KDD*, 2014.

[8] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

[9] Jianfeng Gao, Patrick Pantel, Michael Gamon, Xiaodong He, Li Deng, and Yelong Shen. Modeling interestingness with deep neural networks. In *EMNLP*, 2014.

[10] Lise Getoor and Ben Taskar, editors. *Introduction to Statistical Relational Learning*. The MIT Press, 2007.

[11] Geoff Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *IEEE Sig. Proc. Mag.*, 29:82–97, 2012.

[12] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*, 2013.

[13] B Hutchinson, L. Deng, and D. Yu. Tensor deep stacking networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1944–1957, 2013.

[14] Rodolphe Jenatton, Nicolas Le Roux, Antoine Bordes, and Guillaume Obozinski. A latent factor model for highly multi-relational data. In *NIPS*, 2012.

[15] Charles Kemp, Joshua B Tenenbaum, Thomas L Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, volume 3, page 5, 2006.

[16] Ni Lao, Tom Mitchell, and William W Cohen. Random walk inference and learning in a large scale knowledge base. In *EMNLP*, pages 529–539, 2011.

[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[18] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *ICML*, pages 809–816, 2011.

[19] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing YAGO: scalable machine learning for linked data. In *WWW*, pages 271–280, 2012.

[20] Alberto Paccanaro and Geoffrey E. Hinton. Learning distributed representations of concepts using linear relational embedding. *IEEE Transactions on Knowledge and Data Engineering*, 13(2):232–244, 2001.

[21] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

[22] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Gregoire Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. In *CIKM*, 2014.

[23] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. Learning semantic representations using convolutional neural networks for web search. In *WWW*, pages 373–374, 2014.

[24] Ajit P Singh and Geoffrey J Gordon. Relational learning via collective matrix factorization. In *KDD*, pages 650–658. ACM, 2008.

[25] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. Reasoning With Neural Tensor Networks For Knowledge Base Completion. In *NIPS*, 2013.

[26] Ilya Sutskever, Joshua B Tenenbaum, and Ruslan Salakhutdinov. Modelling relational data using bayesian clustered tensor factorization. In *NIPS*, pages 1821–1828, 2009.

[27] O. Vinyals, Y. Jia, L. Deng, and T. Darrell. Learning with recursive perceptual representations. In *NIPS*, 2012.

[28] Wen-tau Yih, Xiaodong He, and Christopher Meek. Semantic parsing for single-relation question answering. In *ACL*, 2014.

[29] D. Yu, L. Deng, and F. Seide. The deep tensor neural network with applications to large vocabulary speech recognition. *IEEE Trans. Audio, Speech and Language Proc.*, 21(2):388 –396, 2013.

Appendix

Figure 1 and 2 illustrate the relation embeddings learned by DISTMULT and DISTADD using t-SNE. We selected 189 relations in the FB15k-401 dataset. The embeddings learned by DISTMULT nicely reflect the clustering structures among these relations (e.g. /film/release_region is closed to /film/country); whereas the embeddings learned by DISTADD present structure that is harder to interpret.

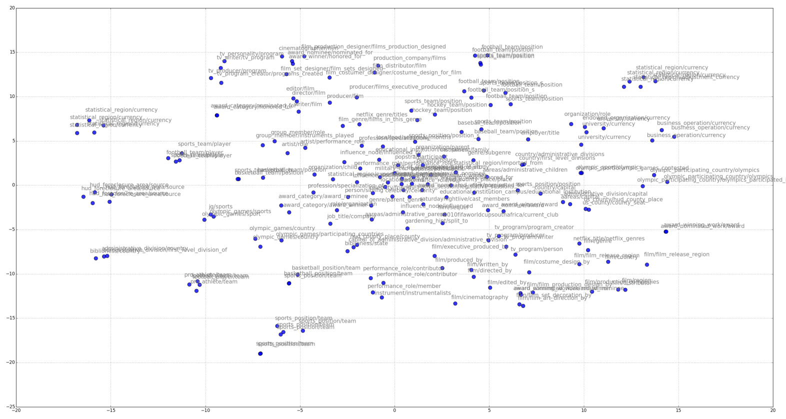

Figure 1: Relation embeddings (DISTADD)

Table 5 shows some concrete examples: top 3 nearest neighbors in the relation space learned by DISTMULT and DISTADD along with the distance values (Frobenius distance between two relation matrices or Euclidean distance between two relation vectors). We can see that the nearest neighbors found by DISTMULT are much more meaningful. DISTADD tends to retrieve irrelevant relations which take in completely different types of arguments.

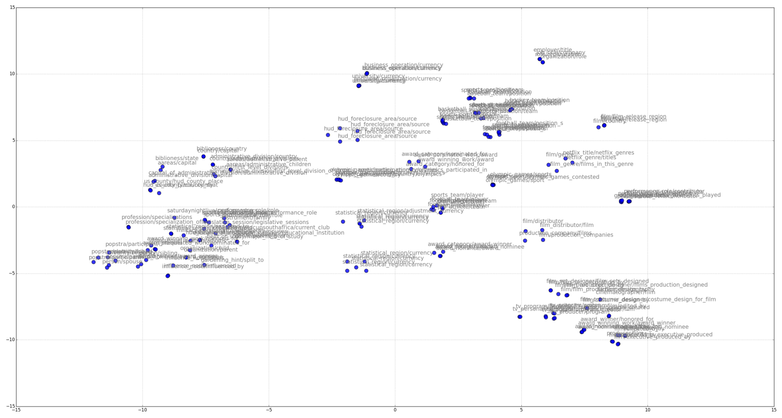

Figure 2: Relation embeddings (DistMult)

	DistMult	DistAdd
/film_distributor/film	/film/distributor (2.0) /production_company/films (3.4) /film/production_companies (3.4)	/production_company/films (2.6) /award_nominee/nominated_for (2.7) /award_winner/honored_for (2.9)
/film/film_set_decoration_by	/film_set_designer/film_sets_designed (2.5) /film/film_art_direction_by (6.8) /film/film_production_design_by (9.6)	/award_nominated_work/award_nominee (2.7) /film/film_art_direction_by (2.7) /award_winning_work/award_winner (2.8)
/organization/leadership/role	/leadership/organization (2.3) /job_title/company (12.5) /business/employment_tenure/title (13.0)	/organization/currency (3.0) /location/containedby (3.0) /university/currency (3.0)
/person/place_of_birth	/location/people_born_here (1.7) /person/places_lived (8.0) /people/marriage/location_of_ceremony (14.0)	/us_county/county_seat (2.6) /administrative_division/capital (2.7) /educational_institution/campuses (2.8)

Table 5: Examples of nearest neighbors of relation embeddings